AFRL-RI-RS-TR-2008-116
Final Technical Report
April 2008

EXTENSIBLE SYSTEMS DYNAMICS FRAMEWORK

University of California

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2008-116 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/

JOHN J. SALERNO
Work Unit Manager

/s/

JOSEPH CAMERA, Chief
Information & Intelligence Exploitation Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
APR 08

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
May 06 – Dec 07

4. TITLE AND SUBTITLE

EXTENSIBLE SYSTEMS DYNAMIC FRAMEWORK

5a. CONTRACT NUMBER
FA8750-06-C-0053

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62702F

6. AUTHOR(S)

Christopher Brooks and Edward A. Lee

5d. PROJECT NUMBER
558S

5e. TASK NUMBER
ES

5f. WORK UNIT NUMBER
DF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The Regents of the University of California, Berkeley
336 Sproul Hall
Berkeley CA 94720-5940

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RIEA
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2008-116

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# WPAFB 08-2621

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Modern defense strategy and execution has become more net-centric and distributed, allowing more information to be made available more rapidly. The warfighter must assemble decision-quality information from potentially inaccurate, or even conflicting, pieces of information collected from multiple sources. The Pedigree Management and Assessment Framework (PMAF) enables the publisher of information to record standard pedigree, such as information about the source, manner of collection, and the chain of modification of that information. In addition, the publisher can define and include other related information relevant to quality assessment, such as domain-specific information about sensor accuracy or organizational structure of agencies. PMAF stores this potentially enormous amount of information in a volume-efficient manner and presents the information to the user in an intuitive graphical format, together with PMAF-generated assessments that allow the user to quickly estimate information quality. PMAF has been created to be compatible with the Air Force Research Laboratory’s Operational Information Management framework and a Web Services environment. It can access pedigree information across communities-of-interest and across network boundaries.

15. SUBJECT TERMS

Ptolemy II, Systems Dynamics, PMESII, National Operational Environment Model

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** John J. Salerno
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	22	**19b. TELEPHONE NUMBER (*Include area code)*** N/A

TABLE OF CONTENTS

1 Introduction

This is the final report of activities in the Extensible Systems Dynamics Framework project performed by the Ptolemy Project, University of California, Berkeley for the Air Force Research Laboratory. This report covers dates from 22-May-2006 through 31-December-2007.

The goal of the Extensible Systems Dynamics Framework Project is to build on top of a pre-existing, open-source modeling framework known as Ptolemy II (see [1] [2] and http://ptolemy.eecs.berkeley.edu) a Commander's Predictive Environment (CPE) that provides a decision and support environment that enables a commander to anticipate and shape the future battlespace. Our target is to support modeling and simulation of large scale dynamics of Political, Military, Economic, Social, Information and Infrastructure (PMESII) systems such as those described in [3]. PMESII is also known as National Operational Environment Modeling (NOEM).Key characteristics of such systems include complex, multiresolution continuous and discrete dynamics and large parameter sets. Our focus will be on facilitating development, maintenance, and evolution of such models, with a particular focus on mechanisms for creating reusable components and mechanisms for scaling the models.

2 Participants

PRINCIPAL INVESTIGATOR:

EDWARD A. LEE (UC BERKELEY, EECS)

GRADUATE STUDENTS:

BEN LICKLY (UC BERKELEY)
ISSAC LIU (UC BERKELEY)
HAIYANG ZHENG (UC BERKELEY)
YE ZHOU (UC BERKELEY)

TECHNICAL STAFF, PROGRAMMER:

CHRISTOPHER BROOKS (UC BERKELEY)

BUSINESS ADMINISTRATOR:

TRACEY RICHARDS (UC BERKELEY)

3 Activities and Findings

Below is a chronology of the primary interactions and shipment of deliverables.

- **July 25, 2006:** Site visit by Berkeley personnel to Rome, NY.
- **October 2, 2006:** Ptolemy II 6.0.alpha online demonstrations, documentation and software released.
- **January 26, 2007:** Video Teleconference between Rome and Berkeley
- **January 14, 2007:** Ptplot 5.6 released
- **February 4, 2007:** Ptolemy II 6.0.2 online demonstrations, documentation and software released.
- **February 12-13, 2007:** The Seventh Biennial Ptolemy Miniconference and tutorial was held in Berkeley, California. Personnel from AFRL attended the tutorial and Miniconference.
- **May 23, 2007:** Site visit by Berkeley personnel to Rome, NY.
- **December 12, 2007:** Site visit by Berkeley personnel to Rome, NY.

During the project period, Berkeley faculty and staff were available for consultation via telephone and email for Ptolemy support questions. In addition, the latest Ptolemy II source code was continuously available to the Rome team via Concurrent Version Control (CVS).

The Berkeley group shipped Ptolemy II 7.0.beta on February 7, 2008. The 7.0.beta release included work performed under this project.

4 Comparison with Statement of work

In this section, we explicitly compare our results with the Statement of Work (SOW) dated February 6, 2006. The title of each section is from the SOW, then the SOW is quoted. Section 4.1 of the 06-February-2006 SOW says:

> “The contractor shall collaboratively develop enhancements, adapt, and incrementally improve performance, extensibility, integration, and usability a pre-existing systems dynamics modeling framework building on an open-source, extensible, open architecture to meet and improve the modeling needs of AFRL/RI researchers.”

This requirement is further detailed in the sections below.

4.1 Simulink Conversion

The SOW says:

> "Convert the Simulink model provided to Ptolemy II and evaluate any limitations that might be present using the model converted."

This requirement was fulfilled by our initial rough conversion of the model from Simulink to Ptolemy II so that we could get a sense of how the model worked. The model was later converted into Ptolemy II by AFRL/RI researchers. During the conversion, we discovered issues surrounding algebraic loops (see below).

The largest issue with Simulink to Ptolemy II conversion is that while the syntax of the model is converted, the semantic differences between the two systems are unknown. A fully operational conversion is outside the scope of this project, it would require developing bridge actors that had similar interfaces as the Simulink blocks.

4.2 Adapt the Ptolemy II framework to ensure a well-suited modeling capability

The SOW says:

> "Adapt the Ptolemy II framework to ensure a well-suited modeling capability."

This requirement is fairly generic. We fulfilled it by:

- Developing new actors and other functionality
- Responding to bug reports and questions promptly.

A list of new functionality and bug fixes is below. Other sections below include more specific enhancements surrounding performance and other areas.

New Actors

- Added Publisher and Subscriber actors. These actors in effect accomplish "wireless" communication for any domain. Wireless connections between components in Ptolemy II allows for cleaner and more modular large models.
- Added SubscriptionAggregator and Triangular actor (both written by AFRL/RIEA, Raymond Cardillo)
- Added the ParameterSet actor, used to read sets of parameters (See 0,"
- Large Parameter Sets," below).
- Added a MovingAverage actor which provides a simpler implementation than a multi-actor based solution. Suggested using an FIR actor in cases where an initial transient is not a problem.
- A new actor called SubMatrix was created. This actor extracts a submatrix from an input matrix.
- Added an ArrayPlotterXY actor.

- We've developed versions of the Publisher actor that include functionality from the Test actor. These actors can be used for regression testing of the code generation version of the model.
- MatrixSplit and MatrixJoin actors were added.

Actor Enhancements and Bug Fixes

- Colt is a Java package used for generating random numbers. As part of this project we made the following enhancements to Colt actors:
    - ColtSeedParameter: Fixed bug where seeds revert to default value
    - All Colt actors (random number generators) are now port parameters. This makes it much easier to use them in a way where the parameters vary dynamically.
    - Responded to problem report by AFRL/RIEA, Raymond Cardillo with ptII.jar and the colt jar files.
- Publisher and Subscriber had the following issues:
    - Publisher and Subscriber: Added tests for converting back and forth between instances and classes. Fixed problems surrounding said conversion.
    - Publisher and Subscriber do not create links if within a class.
    - Publishers now have unique channel names on creation. This solves the class Instantiation issues such as "Instantiating an actor oriented class that contains a Publisher fail". (https://chess.eecs.berkeley.edu/bugzilla/show_bug.cgi?id=66)
    - Fixed problem surrounding deleting a Publisher that was raised by ITT, Jason Smith.
    - Publisher/Subscriber now update links in attributeChanged().
    - Cache the regular expression patterns in SubscriptionAggregator.
    - We also experimented with a different regular expression package (http://www.brics.dk/automaton/) that results in some improvement in performance, but at proportionate increase in memory size.
- The Limiter actor now operates on any Scalar, previously it worked only on doubles.
- The Round actor now throws an exception if the input is NaN.
- FSMActors no longer have a triggerExpression parameter.
- Commutator and Distributor now take a blockSize parameter.
- The Clock actor has been reworked extensively.
- Const actors now can optionally be set to fire for a limited number of times.
- The Sequence actor now has a holdLastValue parameter.
- SetVariable: Fixed bug where failure to evaluate a set expression would cause an infinite sequence of dialogs.
- Scale has been simplified to work like MultiplyDivide
- SequenceToArray now has a multiport output.
- MultiInstanceComposite: Fixed bug where first clone is not getting deleted.
- SequenceToArray.java: Changed output port to multiport to support broadcast.
- NonStrictTest now properly handles cases where we were training and there was no input data.
- LineWriter: Allow the file to be specified at an input port.

- The Display actor now defers opening the window until data arrives
- The MonitorValue actor now clears the display in initialize
- Use FileOrURLAccessor for code reuse with Attributes and Parameters that access files or URLs.
- AddSubtract: Explicitly set type constraints on ports rather than relying on the default behavior.

Other Enhancements or Bug Fixes

- Added StringBufferExec class which allows us to execute arbitrary commands.
- We experimented with the type system in an effort to support array-scalar operations using type polymorphic actors like MultiplyDivide.
- We enhanced ArrayToken and can now specify ArrayToken as an element type.
- Made construction of shared parameter lists much more efficient and improved performance of SharedParameters overall.
- Tuned up Continuous Time (CT) inside Discrete Event (DE).
- Invalidate shadowed settables after name change
- Analyzed ITT, Jason Smith's model which involved many small MoMLChangeRequests in initialize(). Made other suggestions as to performance and style.
- Submitted RFE about ChangeRequests. (https://chess.eecs.berkeley.edu/bugzilla/show_bug.cgi?id=49)
- We fixed a problem concerning composite actors that have no internal actors (the external ports are directly connected).
- Added RecordToken(Map) Constructor and tests so as to make it easier to create RecordTokens from Maps. Added other tests for RecordToken and RecordType.
- We fixed a problem in SubscriptionAggregator that was reported by AFRL/RIEA, Raymond Cardillo. The problem was that copy and paste of a SubscriptionAggregator actor or changing the channel name resulted in the old channels not being deleted. We also added tests of SubscriptionAggregator so as to increase code coverage.
- Worked with AFRL/RIEA, Capt. Robbins on "Create Hierarchy" problem.
- Fixed a long-standing problem with StringToken
- Added upperBound and lowerBound to Accumulator actor, exchanged email with ITT, Brian Hudson about possible improvements in how the Accumulator actor is used in the Integration300 model.
- Fixed a bug in ColtRandomSource where seeds of class instances turn out to be identical. The seed policy for values different from 0L has also changed.
- ColtSeedParameter is now obsolete. We updated the backward compatibility filter accordingly.
- We further refined our solution to the cut and paste problem that occurred when the user selected actors that had parameters that are not present in the pasted location.
- After technical discussions with AFRL/RIEA, Capt. Robbins, we changed how unscheduled actors are reported.
- Fixed problem of missing relations reported by ITT, Jason Smith.

- We improved test coverage of various packages including the actor, data math and moml packages.
- Created OffsetMoMLChangeRequest so that we can offset actors that are pasted and actors that are subclasses or instantiated from classes.
- Fixed problem involving SharedParameters and actor oriented classes. There were two problems. The first was in the clone() method of SharedParameters. The second was where the actor registers as a piggyback with the nearest enclosing opaque composite actor.
- Fixed bug in analysis of SDF models with rate changes.
- Error messages in the SDF Scheduler have been improved.
- Configure now better handles cases where Jython can't be found
- PtDoclet no longer needs to instantiate classes.
- Complex.hashcode(): use bitwise xor here.
- Factored out duplicated code in Effigy classes.
- SDFDirector: Properly deal with actors that are removed: Don't listen to their rate variables anymore.
- The Eclipse installation instructions were updated for Eclipse 3.3.
- SDFScheduler: Improved error messages
- We addressed an issue raised by ITT, Jason Smith surrounding multiports and composite actors.
- MoMLParser was modified to properly handle imports of icons. The problem before was that icons were being repeated in model files.
- Better support for Float and Short in the expression parser
- In Java 1.6, File.toURL() is deprecated. Instead, call toURI().toURL(). See http://java.sun.com/javase/6/docs/api/java/io/File.html#toURL()
- Refactored FilePortParameter and FileParameter to implement a common interface.
- Use PTJAVA_HOME instead of PTJAVA_DIR and avoid problems where java is /usr/bin/java and /usr/bin/java is a link elsewhere.
- Fixes for JNI under Linux: uname -p sometimes is not implemented.
- SRDirector no longer has a period parameter
- SR GuardedCountTimed demo uses an external clock instead of a period
- DE now assumes that all composite actors are strict
- MoMLParser: Fixed subtle bug where there could be two distinct instances of a class definition defined in a MoML file existing in memory at the same time.
- We addressed an issue raised by ITT, Jason Smith surrounding class definitions searching for IDs.
- Created simple filter example for CUBRC, Jared Holsopple.
- After technical discussions with ITT, Brian Hudson, we updated our model to use .+ instead of .* in SubscriptionAggregators.
- Qualified identifiers are allows in the expression language. The two branches of a conditional or expression can now contain arbitrary expressions (even though with lower precedence).

- The expression language is changed so that $(...) in a string can contain a Ptolemy expression, while before this it can only contain an identifier. $... and ${...} are not changed, so they can still only contain identifiers. The attribute operation for graph transformation is now based on this. E.g., if you specify an AttributeOperation with value P+$(Q+R), then in the result of this transformation, the entity's attribute will be set to P+X where X=Q+R. See ptolemy/actor/gt/demo/ConstOptimization for an example.
- We addressed concurrency errors in the SDF kernel that were first reported by AFRL/RIEA, Raymond Cardillo. Here, the problem was that SDFScheduler did not have a clone() method so the _rateVariable field was being shared between multiple models. Gang Zhou, a graduate student at UC Berkeley, identified the problem and implemented the solution.
- We addressed concurrency errors in the graph package. The graph package was not designed for a multithreaded environment. A simple test that executes multiple models resulted in ConcurrentModificationException was created by AFRL/RIEA, Raymond Cardillo and ITT, Brian Hudson. We added a modified version of that test to our nightly build, where the test fails under Solaris with two processors, but passes under Windows with one processor. Our workaround is to make all public methods in TypeLattice synchronized to the class. This workaround could result in slowness.

Software Release Management Improvements

- We fixed thousands of warnings pointed out by FindBugs and Eclipse, which results in a more stable release and improved performance.
- We now have a MIME type for Ptolemy models: vnd.moml+xml. See: http://www.iana.org/assignments/media-types/model/. The MIME type was requested by AFRL/RIEA, Raymond Cardillo so that we can experiment with web services.
- When configure is run, the output is now much cleaner, only "ok" is printed if the test succeeded. To get full output, run "./configure --enable-verbose"

4.2.1 Large Models

The SOW says:

> “Develop the ability to support large models. Address large models by using features in the Ptolemy II environment, such as higher-order components, custom components, and the actor inheritance system to reduce file and memory size. The purpose is to represent complex models compactly and to execute simulations efficiently.”

The model in question is the largest Ptolemy model to date.

In July, 2006, by converting the model to Synchronous Dataflow and making other fixes, we reduced the model size by almost a factor of 5 and the execution time by almost a factor of 10. Many of the performance enhancements help with large models.

4.2.2 Hybrid modeling

> "Develop hybrid modeling capabilities. Identify opportunities to improve simulation performance and model fidelity using mixed modeling capabilities."

We wrote a conference paper [6] that describes how to embed models of computation. The abstract of the paper is reproduced below:

> A model of computation (MoC) is a formal abstraction of execution in a computer. There is a need for composing MoCs in e-science. Kepler, which is based on Ptolemy II, is a scientific workflow environment that allows for MoC composition. This paper explains how MoCs are combined in Kepler and Ptolemy II and analyzes which combinations of MoCs are currently possible and useful. It demonstrates the approach by combining MoCs involving dataflow and finite state machines. The resulting classification should be relevant to other workflow environments wishing to combine multiple MoCs.

On-line resources concerning this paper may be found at
http://www.mygrid.org.uk/wiki/Papers/IccsPaper2007.

4.2.3 Algebraic Loops

> "Algebraic loops. Identify more well-founded ways of accomplishing modeling objectives including explicit modeling of time delays in feedback coupling between portions of the model and analytical solutions to the algebraic equations that are being solved by the Simulink algebraic loop solver."

Partial conversion of the model from Simulink to Ptolemy II helped us with issues surrounding algebraic loops and helped us discuss algebraic loops with the AFRL/RI researchers.

Unification of the Synchronous/Reactive and Continuous Time domains [4] moves Ptolemy II towards a clean semantics which helps us replace algebraic loops.

4.2.4 Large Parameter Sets

> "Large parameter sets. Develop scalable mechanisms that conveniently support large parameter sets and an interface that allows users to set parameters by providing an initialization file. Develop mechanisms for managing model parameters that provides convenient handling of parameter sets."

We enhanced Ptolemy II to have a ParameterSet attribute, which is an attribute that reads multiple values from a file and sets corresponding parameters in the container. This attribute helps multiple runs of a model.

The ParameterSet actor had the following issues:

- ParameterSet now only creates the parameters in the container and reads the data file if the data file name has changed.
- ParameterSet: the $HOME and $CWD now work in the fieldOrURLparameter.
- Improved documentation of ParameterSet. Engaged in discussion about ParameterSet with Kepler users.
- Fixed a ParameterSet problem where Parameter set was not re-reading the parameter file.
- A new version of SharedParameter that uses a static repository to find shared parameters. Do not set the expression if the value of the expression is what has already been inferred from the context.

4.2.5 Command-line interface

> “Command-line interface. Supplement the graphical user interface of Ptolemy II with a command-line interface that allows convenient manipulation of models and parameters in a scripted fashion.”

The command-line interface requirement was deemed as unnecessary. The ParameterSet actor helps obviate the need for a command-lie interface.

4.2.6 Library Maintenance

> “Library maintenance. Design configurations to support the modeling of interest. Support construction of customized model development frameworks that include specialized component libraries and a selected subset of the modeling capabilities.”

The Library maintenance requirement was deemed to be less important than other requirements.

4.2.7 Performance

> “Performance. Identify performance improvements for complex models constructed and when simulation performance proves inadequate.”

In addition to the performance improvement listed in 4.14.1 above, we also addresses issues concerning how long it takes a model to open and run. In September, 2006, we were able to speed up opening by a factor of 8.2 and run time by a factor of 2.8.

In March, 2007, we improved the run-time of the model:

On 3/20, we had the following performance:

```
opening ave time: 163.511 sec.
preinitialize ave time: 953.697 sec., ave mem used: 406640 K
```

```
run ave time: 1087.8 sec., ave mem used: 507497 K
```

On 3/28, after several improvements:

```
opening ave time: 314.967 sec.
run ave time: 76.8122 sec., ave mem used: 543145 K
```

(The preinitialize time is less than 60 seconds so it is not reported.)

Our performance improvements included:

- If the model takes more than 10 seconds to open, then we print statistics.
- We used the FindBugs program to identify and fix hundreds of warnings. The changes result in a decrease of about 1% in total run time of the Integration300.xml model.
- Added Float and Short types, this could help with performance and reduce memory footprint.
- NamedObj.getAttribute() was tuned, which resulted in a 2-3% speed up in mini-model-aggregator.xml.
- IntToken._value is now final; this resulted in roughly a 1% speed up in mini-model-aggregator.xml.
- How we access the unit system was modified so that we avoid an excessive number of method calls to check to see if we need to check the units.
- Updated Publisher and Subscriber to fix bug when these actors are used in external class files. Publisher was also modified so the second run is faster.
- Only call validateSettables() and validate() when necessary (improves performance).
- The validate() method now returns a Collection, which can be used to avoid duplicate calculations (improves performance).
- SDFScheduler now uses a HashSet instead of a Treeset.
- SDFScheduler no longer invalidates the schedule if the parameter values have not changed.
- IORelation does not get write access if we are not going to write.
- Added TemporaryVariable, which helps us avoid incrementing the Workspace version. Changes to other classes make rerunning models very fast because we avoid type checking a model again.
- The attributeChanged() method of several actors was adjusted so as to avoid being called between runs.
- The array typing mechanism was changed.
- NamedObj.instantiate() and ComponentEntity.instantiate() were tuned.

We also used code generation [7] to convert the model to C code, reducing the run time from 140 seconds to 3 seconds. Much of this work involved tracking down numeric differences in the output between the regular interpreted run and the code generation run.

Code Generation Specific Changes

- Added C Code Generator support for MovingAverage, VectorAssembler, VectorDisassembler, ColtBinomial, ColtPoisson, ColtBinomialSelector, WallClockTime, Publisher, and Subscriber.

- C Code Generator script now generates code for models that have plotters. Formerly, the user was required to run Vergil to generate code for models that use plotters.
- Added C Code Generator support for double matrices.
- The code generator can now generate code coverage statistics for C code.
- Added support of C longs and unsigned bytes to code generator.
- Code generation version of Expression now handles "time" and "iteration" keywords properly.
- EmbeddedCActor has been tuned up. This actor now allows us to run arbitrary C code from within Java and then generates the entire model in C and use the same code.
- Code generator can now optionally display runtime data.
- The getHeaders() method in code generator wrappers now properly calls the super class.
- Added fileDependency block to codegen that allows us to copy files to the codegen build directory.
- Fixed codegen Matrix bug involving ; and ,
- Codegen matrices now have add and subtract
- Codegen MultiplyDivide fixed to correctly handle token data type
- The Butterfly demo is now a better codegen demo
- Code generator has improved error messages
- EmbeddedCActor has better error handling
- PlotterBase has a different prompt to exit
- Pad buffers even when the buffer size divides read tokens and write tokens. Remove accessor methods for public parameters in CodeGenerator.
- Support for the PN domain is in the process of being implemented.
- Tests for problems in array arithmetic have been added.
- Introduction of two new code generation parameters:
    - Optionally turn buffer padding off.
    - Optionally allow dynamic referencing of multiports in c code.
- Allow multiports in EmbeddedCActor.
- Use code generation for value transfer of FSMActor ports.
- Fixed getSize(String) to fetch variable from the model scope instead of fetching the attribute only from the component.
- Took out extra white spaces in the generated code for SumNode.
- Added assignment statements "_childCode = result" in a couple places to properly propagate the generated string.
- Added the "TokenFireBlock" code block. This handles nested array input type. The check for input array type is taken out. The input array type is implicitly assumed.
- Added codegen support for MultiInstanceComposite. MultiInstanceComposite creates a parameterized number of instances of a composite actor. This actor could be a big help in the model under development.
- Refactored code generator so that operations on tokens will be in one place instead of spread throughout the actors.

- The Code Generator is better at handling operations on numerics in a polymorphic fashion.
- Implemented type-polymorphic functions (i.e. add, subtract, negate, convert) that can be statically refined (i.e. add_Int_Double). These functions can be called within a code block using the $ macro syntax, such as "$add_Int_Double(a1, a2)".
- Files containing c code implementations (as code blocks) of these refined functions are in the codegen/c/kernel/type/polymorphic directory. C CodeGenerator.generateType-ConvertCode() automatically appends the referenced function implementation into the generated code.
- CompiledCompositeActor: Better error messages if we have class version problems.
- The ability to interface the code generator to C++ is under development (Funded by another project)
- The code generator can handle Variables so that we can someday support code generation of the SetVariable actor.
- The ColtBinomialSelector actor now handles outputs that are not connected. A problem where the code generation version was not resetting the "selected" variable to 0 was also fixed. Test cases were added for both issues
- Compiliation of the C file was taking 39 minutes because the initialize() and other methods had bodies with up to 15k lines.
- Fixed bugs in several actors where we used $val instead of $ref.
- The shift operand ("<<") now works with variable references.
- ColtPoisson now properly handles inputs that are Infinity.
- The C version of the Round actor was using rint() instead of round()!
- The Code Generator refactoring work is mostly complete, though there are still failing tests.
- Support Publishers that have no connections. We now print a warning message.
- PublisherTest handles multiports

4.2.8 Ease of Use Improvements

> Ease of use improvements. Develop ease-of-use improvements that mitigate awkwardness or other difficulties in the user interface.

We updated The Vergil user interface to include Relation Groups. The Vergil documentation describes this feature:

> “Relations mediate connections between ports. For flexibility, particularly with visual syntaxes, the Ptolemy II abstract syntax permits any number of relations to be involved in any one connection. Relations may be linked to other relations. Any two relations that are linked are said to be members of the same relation group. Specifically, a relation group is a maximal set of linked relations. Semantically, a relation group has the same meaning as a single relation. The API

of the Relation class, support linking and unlinking relations, and also provides a method to obtain a list of all the relations in a relation group."

"In a relation group, there is no significance to the order in which relations are linked, unlike the order in which ports are linked to relations. Also, unlike links between relations and ports, there is no significance to multiple links between the same relations. Any two relations are either linked or not linked."

User Interface Changes

- The documentation can now be created if there are spaces in the $PTII path.
- Viewing Documentation now works within the Case actor.
- At Raymond Cardillo's request, we added an "Open Instance" menu item.
- Fixed Multiport drag bug (https://chess.eecs.berkeley.edu/bugzilla/show_bug.cgi?id=13)
- Fixed bug involving underscores and cut and paste (https://chess.eecs.berkeley.edu/bugzilla/show_bug.cgi?id=61)
- Improved Port Parameter drag group problem, though the problem is not completely solved. (https://chess.eecs.berkeley.edu/bugzilla/show_bug.cgi?id=34)
- Highlight actors that cause exceptions. Now, when an exception occurs, the cause of the exception and all its containers will be highlighted in Vergil. The highlight is cleared when you successfully re-run the model. (https://chess.eecs.berkeley.edu/bugzilla/show_bug.cgi?id=30)
- Provide a DependencyHighlighter attribute in the Utilities menu. We have put into the Utilities menu an attribute that you can drop on any actor that adds four commands to its context menu: highlight dependents, clear dependents, highlight prerequisites, and clear prerequisites. This seems potentially generically useful enough to have in the library.
- Further work on the testing of instantiating actor oriented (AO) classes with publishers. We have replicated a bug where instantiating an AO class that contains a Publisher throws an exception. (https://chess.eecs.berkeley.edu/bugzilla/show_bug.cgi?id=66)
- The "about:" facility now reports problems with the size of TypedComposite actors that are used by models. We need to fix these warnings.
- The font size has been increased after we observed that Dr. Romeau found clicking on HTML links to be difficult. The small font size has been a problem for awhile.
- Fixed problem with FSM actions where clearing the value did not persist. (https://chess.eecs.berkeley.edu/bugzilla/show_bug.cgi?id=12)
- In SaveAs, "save submodel only" now works for modal models (https://chess.eecs.berkeley.edu/bugzilla/show_bug.cgi?id=53)
- Added explicit test for liberal links. (https://chess.eecs.berkeley.edu/bugzilla/show_bug.cgi?id=26)
- FSM viewer now has automatic layout. (https://chess.eecs.berkeley.edu/bugzilla/show_bug.cgi?id=7)
- We can now build the Ptolemy documentation from within Eclipse. (https://chess.eecs.berkeley.edu/bugzilla/show_bug.cgi?id=35)

- Fixed copy and paste problems with parameters. (https://chess.eecs.berkeley.edu/bugzilla/show_bug.cgi?id=21)
- Fixed problem with copy and paste of actor with multiports.
- Fixed problems with PortParameter icon rendering. (https://chess.eecs.berkeley.edu/bugzilla/show_bug.cgi?id=34)
- Modified the user interface so that if a window is off screen, and then place it on screen.
- Actors in the left hand actor pane now have a 'Get Documentation' menu choice.
- Fixed bug where opening view text on an unsaved model would cause the unsaved model not to be saved on close. (https://chess.eecs.berkeley.edu/bugzilla/show_bug.cgi?id=67)
- CreateHierarchy now deletes entities before deleting properties. This helps avoid deleting properties such as top level parameters upon which the entities depend.
- Formerly, if the user opens up a composite actor and then runs the top level and there is an error, then the composite actor window pops up with the error message. Now, the current window (the top level) should stay up.
- The FSM UI has been updated
  - The initial state is now shown in bold.
  - States are shown in rounded boxes that fit the name.
  - The initial state is selected at the state rather than in the Background.
  - The first state inserted is by default the initial state.
  - States with refinements are visually different from states without.
  - Final states in FSMs now have double boxes.
  - Resetting transitions in FSMs now have unfilled arrows]
  - PtolemyQuery and FSM Transitions now supports text height hints
  - FSM now supports rendering preemptive transitions.
  - FSM supports annotations on transitions.
- Discussed double click action for looking inside actors
- Analyzed screen update problems with Java 1.5.0_06.

4.3 Software Delivery

> “Deliver all computer software developed or assembled to be completely maintainable and modifiable without reliance on any non-delivered computer programs or documentation in accordance with the Contract schedule and the following. Deliver software acquired and included as a component in the software developed and delivered.”

Our nightly build system and copyright management system identified any dependencies on third party software. Dependencies are clearly documented in the installation notes. Ptolemy II includes many third party packages, to our knowledge; all third party software used by Ptolemy II for this project is included in the release.

4.3.1 Y2K

> “All information technology items must be Year 2000 compliant or non-compliant items must be upgraded at no additional cost to be Year 2000 compliant. Year 2000 compliant means information technology that accurately processes data/time data (including, but not limited to calculating, comparing, and sequencing) from, into, and between the twentieth and twenty-first centuries, and the years 1999 and 2000 and leap year calculations. Furthermore, Year 2000 compliant information technology, when used in combination with other information technology, shall accurately process date/time data if the other information technology properly exchanges date/time data within it.

Ptolemy II is implemented in Java and requires Java 1.5 to compile and run. Java 1.2 and later releases are [5] Y2K compliant.

4.3.2 Software Delivery

> “Deliver all computer software developed under this effort as source and object (executable) code. Include the commented source listings and source coded for the target computer system. (CDRL A003)”

All software was and is available for download via the Ptolemy CVS repository, see http://chess.eecs.berkeley.edu/ptexternal.
As a part of this project, we set up a build that creates tar files of our source code on a nightly basis. We shipped Ptolemy II 6.0 and 7.0.beta as full Ptolemy releases with source and object code.

4.3.2.1 Licenses of software used by Deliverables

> “Deliverables must not inhibit re-use or redistribution. AFRL must be able to use the results of this effort (software and concepts) to develop derivative products in the future without additional licensing costs or distribution restrictions. Therefore, deliverables must not require any dependencies that extend, adapt, or introduce additional licensing or redistribution terms.”

The software was distributed with the BSD-Style Ptolemy II Copyright.

```
Copyright © 1990-2008 The Regents of the University of California. All
rights reserved.
Permission is hereby granted, without written agreement and without
license or royalty fees, to use, copy, modify, and distribute this
software and its documentation for any purpose, provided that the above
copyright notice and the following two paragraphs appear in all copies
of this software.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY
FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES
ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF
```

4.3.2.2 Freely available Java Technologies

> “Java technologies that are freely available, freely redistributable, and platform independent, must be used to develop any of the deliverables that provide functionality specified by the contract requirements.”

Ptolemy II is developed using Java 1.5 and Java 1.6.

Software Documentation
“Develop complete software documentation to include installation, user, and maintenance instructions as appropriate for the application. (CDRL A004)”

The documentation for Ptolemy II 6.0 is available from
http://ptolemy.eecs.berkeley.edu/ptolemyII/designdoc.htm
The documentation for Ptolemy II 7.0 will be available in March, 2008.

5 Conclusions

As shown by the changes and performance improvements above, the AFRL/IF and Berkeley researchers made great progress towards solving the National Operational Environment Modeling (NOEM) problem. Ptolemy II has been successfully enhanced to handle large models with large parameter sets in an efficient manner. Further work in the area of code generation, usability, configuration management and performance tuning is warranted.

6 References

[1] C. Brooks, E.A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, H. Zheng (eds.), "**Heterogeneous Concurrent Modeling and Design in Java (Volume 1: Introduction to Ptolemy II),**" EECS Department, University of California, Berkeley, UCB/EECS-2007-7, January 11, 2007. (available at **http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-7.html**)

[2] Johan Eker, Jörn W. Janneck, Edward A. Lee, Jie Liu, Xiaojun Liu, Jozsef Ludvig, Stephen Neuendorffer, Sonia Sachs, Yuhong Xiong, "Taming Heterogeneity---the Ptolemy Approach," *Proceedings of the IEEE*, v.91, No. 2, January 2003. (available at http://www.ptolemy.eecs.berkeley.edu/publications/papers/03/TamingHeterogeneity)

[3] M. J. Robbins, "Investigating the Complexities of Nationbuilding: A Sub-National Regional Perspective," Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, Thesis, AFIT/GOR/ENS/05-16, March 2005

[4] Haiyang Zheng, "Operational Semantics of Hybrid Systems", EECS Department, University of California, Berkeley, Technical Report No. UCB/EECS-2007-68, May 18, 2007. (Available at http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-68.html)

[5] Ian F. Darwin, “Java Cookbook,” p149, O’Reilly, 2001, ISBN 0596001703.

[6] Antoon Goderis, Christopher Brooks, lkay Altintas, Edward A. Lee, "Composing Different Models of Computation in Ptolemy II and Kepler," 2007 Proceedings, International Conference on Computational Science (ICCS), May, 2007 (Available at http://chess.eecs.berkeley.edu/pubs/193.html)

[7] Gang Zhou, Man-Kit Leung and Edward A. Lee, "A Code Generation Framework for Actor-Oriented Models with Partial Evaluation," EECS Department, University of California, Berkeley, Technical Report No. UCB/EECS-2007-29, February 23, 2007.

List of Abbreviations and Acronyms

Colt A set of random number libraries from http://dsd.lbl.gov/~hoschek/colt/

CT Continuous Time. The Continuous Time model of computation aims to help the design of systems that have continuous dynamic, for example, analog circuits, mechanical systems, and the continuous environment for embedded systems.

CVS Concurrent Version Control System. CVS is uses to manage the Ptolemy II source tree.

DE Discrete Event. The Discrete Event model of computation provides a general environment for time-oriented simulations of systems such as queueing systems, communication networks, and hardware systems.

FIR Finite Impulse Response - A type of signal processing filter.

FSM Finite State Machine. The Finite State Machine model of computation consists of a set of finite states, transitions and actions.

Kepler A project that uses Ptolemy for scientific workflows (http://www.kepler-project.org)

MIME Multipurpose Internet Mail Extensions

MoC Model of Computation. The rules that govern the interaction, communication, and control flow of a set of components.

NOEM National Operational Environment Modeling

SDF Synchronous Data Flow. Synchronous dataflow (SDF) is a special case of dataflow where the flow of control is sufficiently regular to be completely predictable at compile time.

SROM Stabilization and Reconstruction Operations Model

PMESII Political, Military, Economic, Social, Information, and Infrastructure